



UNIVERSITÀ DEGLI STUDI DELL'AQUILA

**DIPARTIMENTO DI
INGEGNERIA E SCIENZE
DELL'INFORMAZIONE E MATEMATICA**



**CORSO DI LAUREA IN
INGEGNERIA INFORMATICA E AUTOMATICA**

**Sviluppo di un sistema di stima dell'assetto per
un'applicazione di localizzazione indoor**

Relatore:

Prof. Luigi Pomante

Candidato:

Fabio Di Sabatino

Co-relatori:

Giacomo Valente, Marco Santic

Matricola:

246526

ANNO ACCADEMICO 2017–2018

Indice

Introduzione	1
1 Contesto applicativo	4
1.1 Context-aware computing	4
1.2 Indoor Positioning System Service	6
1.3 Stima della posizione	8
1.3.1 Range based	9
1.3.1.1 Received Signal Strength Indicator - RSSI	9
1.3.1.2 Time Of Arrival measurements	11
1.3.1.3 Time Difference Of Arrival	12
1.3.2 Angle Based	13
1.3.2.1 Angle of Arrival	13
1.4 Tecniche di localizzazione	15
1.4.1 MIN-MAX	15
1.4.2 Trilaterazione	16
1.4.3 Triangolazione	17
1.5 Sensor Data Fusion	18
2 Descrizione del lavoro	20
2.1 Livello black box	21
2.2 Livello sottosistemi	22
2.3 Livello moduli	25
3 Unità di misura inerziale	29
3.1 Accelerometro	29
3.2 Giroscopio	31
3.3 Magnetometro	32
3.4 Modello di misura	34

4	Stima dell'assetto tramite sensor fusion	37
4.1	Rappresentazione geometrica dell'assetto di un corpo rigido nello spazio	37
4.1.1	Angoli di Eulero	39
4.1.2	Quaternioni	39
4.2	Sensor fusion mediante un filtro di Kalman	39
5	Implementazione	40
5.1	Modalità di computazione	40
5.1.1	Low Computation mode	40
5.1.2	High Computation mode	40
5.1.3	Testing Computation mode	40
5.2	Canali di comunicazione	40
5.2.1	I2C	40
5.2.2	USB CDC	40
6	Analisi e validazione dei risultati	41
6.1	Analisi temporale	41
6.1.1	Analisi del tempo di lettura dei dati grezzi dall'IMU	42
6.1.2	Analisi del tempo di trasmissione dei dati tramite USB . . .	43
6.1.3	Conclusioni analisi temporale	44
7	Conclusioni e prospettive future	47
A	Quaternioni e rotazioni nello spazio	48
B	Filtro di Kalman	49
C	Script	50
C.1	Testing	50
C.1.1	Script per la stima del tempo di lettura di dati provenienti dall'IMU	50
C.1.2	Script per la stima del tempo di trasmissione di dati al modulo App	51
	Bibliografia	52

Elenco delle figure

1	Planimetria originale	1
2	Planimetria alterata	1
3	Rete allo step 1 dell'esploratore	2
4	Rete allo step 1 dell'esploratore	2
5	Rete allo step 3 dell'esploratore	2
6	Rete allo step 4 dell'esploratore	2
7	Rete allo step 5 dell'esploratore	3
8	Rete allo step 6 dell'esploratore	3
1.1	Esempio esplicativo del concetto di contesto	5
1.2	Sondaggio tra 74 casi di studio di applicazioni IPS [8]	7
1.3	RSSI - Andamento della potenza in funzione della distanza percorsa dal segnale	9
1.4	ToA - Principio di funzionamento	11
1.5	TDoA - Principio di funzionamento	12
1.6	TDoA - Stima della posizione lungo l'iperbole identificata da due nodi	13
1.7	AoA - Stima della posizione attraverso l'angolo di incidenza	14
1.8	MIN-MAX - Tecnica di posizionamento	16
1.9	Trilaterazione - Esempio esplicativo	16
1.10	Triangolazione - Esempio esplicativo	18
2.1	Rappresentazione dei livelli d'astrazione utilizzati per descrivere il sistema	20
2.2	Rappresentazione del sistema come black box	21
2.3	Ambiguità nella georeferenziazione di un secondo nodo utilizzando soltanto la distanza dal precedente	23

2.4	Rappresentazione dei sottosistemi necessari per georeferenziare i nodi della rete	24
2.5	Rappresentazione dei sottosistemi in moduli	25
2.6	Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 1	26
2.7	Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 2	27
2.8	Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 3	27
3.1	Rappresentazione esemplificativa di un accelerometro capacitivo a riposo	30
3.2	Rappresentazione esemplificativa di un accelerometro capacitivo che subisce una forza esterna	31
3.3	Rappresentazione di un giroscopio vibrante per la misura della velocità angolare lungo l'asse z	32
3.4	Rappresentazione esemplificativa di un magnetometro capacitivo lungo l'asse z	33
3.5	in 3.5(a) il b -frame nell'istante t_1 e t_2 relativamente al n -frame, in 3.5(b) l' n -frame in latitudine φ e longitudine λ , l' e -frame all'angolo $\alpha(t) = \omega_{ie}t$ e l' i -frame	35
6.1	In 6.1(a) le operazioni basilari per la modalità LCM , in 6.1(b) le operazioni basilari per le modalità HCM e TCM	42

Introduzione

Il lavoro di questa tesi si colloca in un progetto finalizzato alla realizzazione di un sistema di geolocalizzazione di operatori in contesti privi di segnale GPS. Quest'ultimo permetterà ad un "esploratore" di creare dinamicamente una rete di nodi all'interno di zone nelle quali il segnale GPS è assente o comunque debole. Questa rete verrà poi ampliata ed utilizzata dagli operatori successivi ad esso per geolocalizzarsi all'interno della zona ed intervenire in maniera ottimale.

Uno scenario esemplificativo è quello di un vigile del fuoco che interviene in un edificio per soccorrere una persona. Tale operatore è considerato un esploratore, poiché è il primo ad intervenire e la planimetria dell'edificio risulta essere ignota e/o cambiata a seguito dell'evento disastroso (Fig.1 e Fig.2).

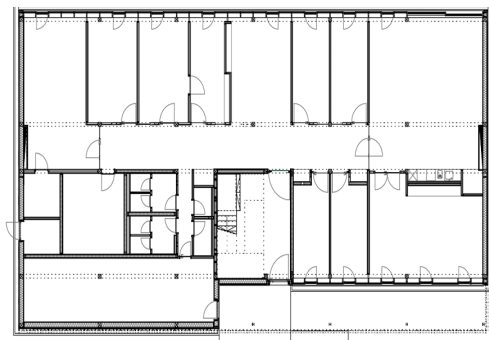


Figura 1: Planimetria originale

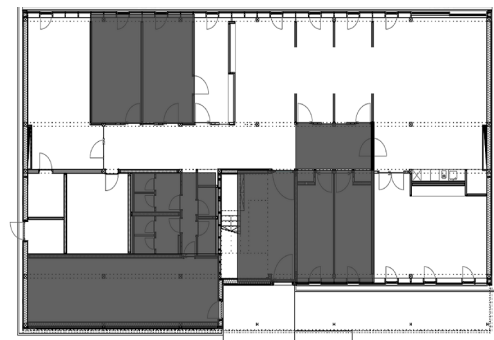


Figura 2: Planimetria alterata

I rettangoli in grigio (Fig.2) rappresentano aree non più accessibili dell'edificio, mentre le mura interrotte nuovi percorsi creati a causa dei crolli. Durante tutta la fase di *scouting* (esplorazione, ricerca, indagine), l'esploratore posizionerà un'ancora (nodo) ogni 20 metri approssimativamente e ogni qualvolta la precedente risulti non essere più in line-of-sight (linea visiva). Così facendo si "lascerà dietro una scia di briciole" che gli permetteranno di orientarsi all'interno dell'edificio, di eseguire il percorso all'inverso o di ricevere supporto da un'ulteriore operatore. Osservando dalla Fig.3 alla Fig.8, si può notare come la rete cresca man mano che l'esploratore avanza all'interno dell'edificio.

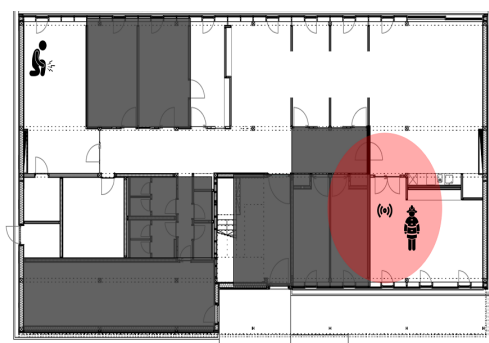


Figura 3: Rete allo step 1 dell'esploratore

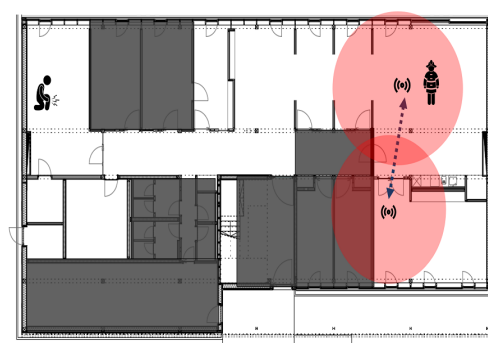


Figura 4: Rete allo step 1 dell'esploratore

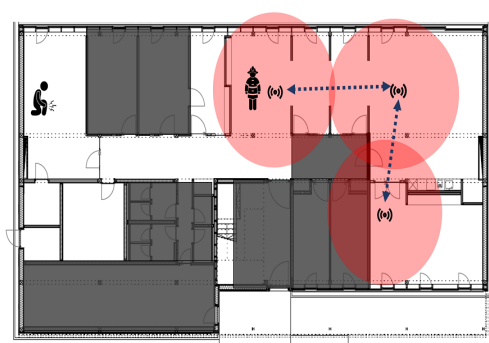


Figura 5: Rete allo step 3 dell'esploratore

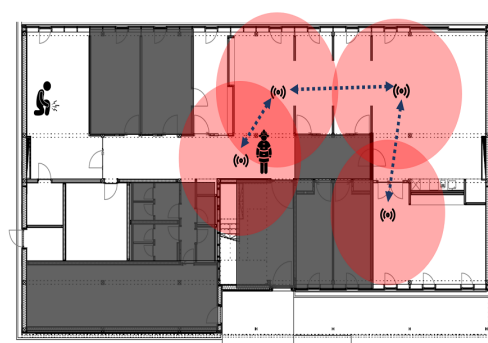


Figura 6: Rete allo step 4 dell'esploratore

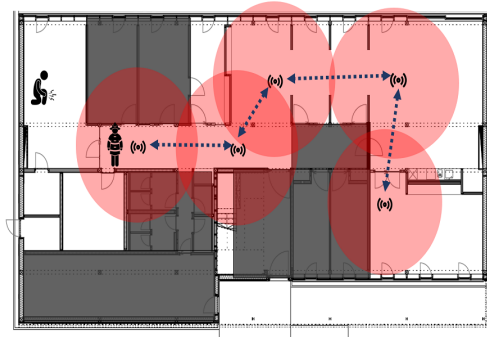


Figura 7: Rete allo step 5 dell'esploratore

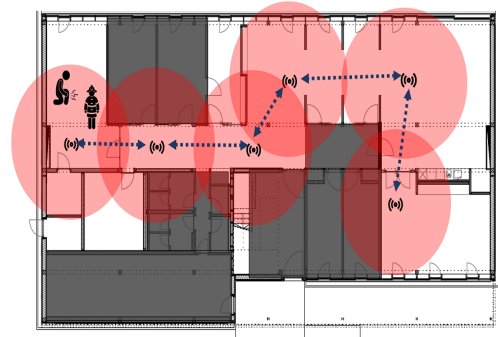


Figura 8: Rete allo step 6 dell'esploratore

Una volta creata l'infrastruttura, gli operatori potranno comunicare e condividere informazioni come posizione e stato.

La presente tesi è così strutturata:

Capitolo 1: Viene descritto il problema della geolocalizzazione indoor e lo stato dell'arte.

Capitolo 2: Viene descritto il lavoro svolto con un approccio top down.

Capitolo 3: Vengono illustrate le tecnologie utilizzate (sensori MEMS e UWB) e le problematiche legate ad esse.

Capitolo 4: Viene descritto l'elaborazione dei dati grezzi e l'algoritmo di data fusion utilizzato.

Capitolo 5: Si riporta l'implementazione software e hardware dei sottosistemi.

Capitolo 6: Si analizzano i risultati ottenuti.

Capitolo 1

Contesto applicativo

Nella parte iniziale di questo capitolo viene spiegato il contesto applicativo di questa tesi, una volta delineato vengono illustrate le tecniche basilari più utilizzate in questo ambito.

1.1 Context-aware computing

Il sistema realizzato nell'ambito di questa tesi, si colloca nel paradigma di computazione noto come *Context-aware computing*, ovvero un'applicazione nel quale i servizi utilizzano informazioni relative al contesto. In [1] si definisce come *context*:

Ogni informazione che può essere usata per caratterizzare la situazione di un'entità. Ovvero una persona, un posto o un oggetto che è considerato rilevante all'interazione tra l'utente e l'applicazione, inclusi quest'ultimi.

Questa definizione facilita il lavoro di progettazione e sviluppo di un'applicazione permettendo di identificare quali informazioni sono importanti e quali no. Si consideri un'applicazione nel quale l'utente deve registrare il peso degli oggetti presenti nel magazzino tramite una bilancia come mostrato dalla Fig.1.1:



Figura 1.1: Esempio esplicativo del concetto di contesto

Nello scenario descritto le *entità* sono rispettivamente utente e sistema, mentre due possibili informazioni riguardanti il contesto sono la presenza di altre persone e la posizione geografica del magazzino.

La presenza di altre persone nelle vicinanze non influisce il compito dell'utente, quindi non può essere considerato come informazione contestuale. La posizione geografica del magazzino invece sì, infatti se quest'ultimo fosse situato in Italia il peso verrebbe calcolato in chilogrammi mentre se fosse situato negli USA verrebbe calcolato in libbre.

I sistemi che reperiscono, usano o interpretano queste informazioni contestuali sono detti *context-aware* e vengono definiti in [1] come:

Un sistema è context-aware se usa il contesto per fornire informazioni rilevanti e/o servizi agli utenti, dove la rilevanza dipende dal compito degli utenti.

Uno dei tipi di *context-aware* più utilizzati si basa sul contesto della localizzazione, ovvero servizi basati sulla conoscenza di dove qualcosa o qualcuno si trovi. Nell'era moderna i servizi basati sulla localizzazione (in inglese: Location based services LBSs) stanno assumendo sempre più importanza nelle attività quotidiane dell'uomo grazie alle molteplici possibili applicazioni, tra le quali navigazione assistita per autoveicoli, tracking di persone sensibili (bambini, anziani, malati), servizi di emergenza e così via.

I LBSs vengono divisi in due macro categorie:

- **OPSs:** Outdoor Positioning System Service, ovvero servizi di localizzazione in ambienti aperti.
- **IPSs:** Indoor Positioning System Service, ovvero servizi di localizzazione in ambienti indoor.

La tecnologia satellitare nota come Global Positioning System (GPS) è la tecnologia dominante negli OPSs. Attraverso una rete dedicata di satelliti artificiali in orbita, fornisce ad un terminale mobile o ricevitore GPS informazioni sulle sue coordinate geografiche ed orario, in ogni condizione meteorologica, ovunque sulla Terra o nelle sue immediate vicinanze ove vi sia un contatto privo di ostacoli con almeno quattro satelliti del sistema. La localizzazione avviene tramite la trasmissione di un segnale radio da parte di ciascun satellite e l'elaborazione dei segnali ricevuti da parte del ricevitore [2].

Il grande limite di questa tecnologia è che i ricevitori devono essere nella line of sight (letteralmente a vista d'occhio) di almeno quattro satelliti nel cielo, questo significa che all'interno di edifici e spazi chiusi il segnale viene attenuato e i sistemi perdono di accuratezza. Quindi la tecnologia GPS non è adatta ai servizi di localizzazione indoor.

Il sistema realizzato nell'ambito di questa tesi si colloca nell'ambito degli IPSs approfonditi nel paragrafo successivo.

1.2 Indoor Positioning System Service

Un sistema di posizionamento indoor (in inglese: Indoor positioning system o IPS) è un sistema in grado di localizzare *oggetti* o *persone* all'interno di edifici utilizzando onde radio, campi magnetici, segnali acustici e/o altre informazioni raccolte dai sensori all'interno di dispositivi mobili [3] o da altri appositamente installati nell'ambiente. Questi sono una specializzazione dei più generici sistemi **RTLS**, standardizzati dall'*International Organization for Standardization and the International Electro Technical Commission* (ISO/IEC 24730). Lo standard definisce i sistemi RTLS come:

“ I Real time locating system sono sistemi wireless con l'abilità di localizzare la posizione di oggetti ovunque essi siano in uno spazio definito in un certo momento che è, o si avvicina, real time. La posizione è derivata dalla misurazione delle proprietà fisiche del collegamento radio.”

La differenza tra RTLS e IPS è che i primi sono stati pensati per le compagnie che vogliono tracciare i propri oggetti e le persone, fornendo uno storico di dove sono stati e dove si trovano ora, mentre gli IPS sono pensati per essere utilizzati da utenti su dispositivi mobili per navigare ed orientarsi all'interno di edifici. Come già accennato, gli IPS [4] permettono di creare una vasta gamma di servizi, ad esempio:

- **Way-Finding:** permettere di navigare in edifici complessi, come ad esempio aeroporti, seguendo il percorso indicato.
- **Ricerca dei punti d'interesse,** aumentare la customer experience facendo trovare all'utente ciò che desidera.
- **Multi-Dot:** visualizzare in una mappa le posizioni degli utenti per tracciare persone potenzialmente in pericolo (bambini, anziani).
- **Marketing di prossimità:** realizzare marketing mirati, inviando annunci sulle ultime offerte.

L'elenco di cui sopra rappresenta solo un ridotto sottoinsieme dei potenziali campi applicativi (vedi Fig.1.2), per questo motivo negli ultimi anni [7] l'interesse nella ricerca e nello sviluppo di sistemi di questo tipo è cresciuto sempre più tra le aziende, che hanno percepito la possibilità di grandi profitti in un mercato non ancora esplorato del tutto.

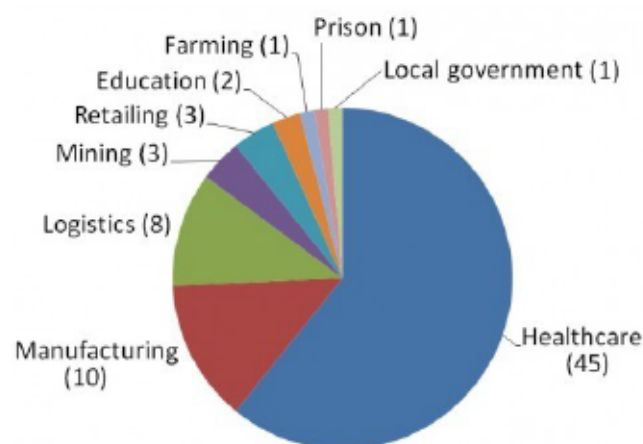


Figura 1.2: Sondaggio tra 74 casi di studio di applicazioni IPS [8]

Secondo un sondaggio di *Markets and Markets* e un articolo pubblicato da *The International News Magazine*, il mercato degli IPS subirà una crescita annuale media del 42.1% arrivando a valere 2.60 bilioni di dollari nel 2018. Questo dà un'idea del perché grandi aziende come Google, Sony, Microsoft e Apple stiano investendo in questo settore.

Un'ulteriore spinta è data dal fatto che, al contrario del GPS per gli OPS (vedi 1.1), tuttora non esiste uno standard di riferimento per gli IPS. Infatti sul mercato sono disponibili diversi tipi di IPS commerciali che si differenziano in base al principio di funzionamento e alle tecniche utilizzate, utilizzando hardware specifico o la combinazione di più sistemi.

1.3 Stima della posizione

Gli IPS possono essere classificati sulla base di diversi fattori, uno di questi è su come determinano la posizione dei.

Stimare [4] la distanza tra dispositivi wireless è utile perché attraverso questa informazione è possibile determinarne (con un certo errore) la posizione di un ricevitore rispetto ad un trasmettitore ([5], [6]), queste tecniche si distinguono in:

- **Range based:**

- *RSSI* - potenza del segnale radio ricevuto (sez.1.3.1.1)
- *ToA* - tempo d'arrivo: (sez.1.3.1.2)
- *TDoA* - differenze del tempo di arrivo (sez.1.3.1.3);

- **Angle based:**

- *AoA* - Angle of Arrival (sez.1.3.2.1).

Per poter determinare le distanze si devono distinguere i punti di riferimento (che hanno delle coordinate note) dai nodi senza posizione nota a cui assegnare delle coordinate. Si dicono:

- **Anchor:** i nodi le cui coordinate sono note
- **Target:** il nodo di cui non si conosce la posizione.

L'obiettivo del posizionamento è assegnare le giuste coordinate agli Unknown rispetto ad un sistema di riferimento. Questo è strettamente legato all'implementazione dell'IPS e così anche la codifica delle posizioni all'interno del sistema di riferimento, le coordinate potrebbero essere restituite all'utente in maniera relativa ("vicino alla cucina") oppure assoluta ("tre metri in direzione ovest dal nodo 1").

1.3.1 Range based

Nel posizionamento dei nodi basato sulla distanza la stima della posizione del target dipende dai seguenti parametri:

- il tempo trascorso tra l'emissione e la ricezione del segnale radio;
- la distanza euclidea tra ogni emettitore ed il ricevitore;
- la potenza del segnale ricevuto.

In alcuni casi sono necessarie tre o più Anchor per ottenere le coordinate da assegnare allo Unknown.

1.3.1.1 Received Signal Strength Indicator - RSSI

La comunicazione [4] tra dispositivi wireless (senza fili) avviene tramite lo scambio di segnali propagati nell'aria. Durante la propagazione i segnali tendono ad attenuarsi con l'aumentare della distanza percorsa fino a non essere più percepibili.

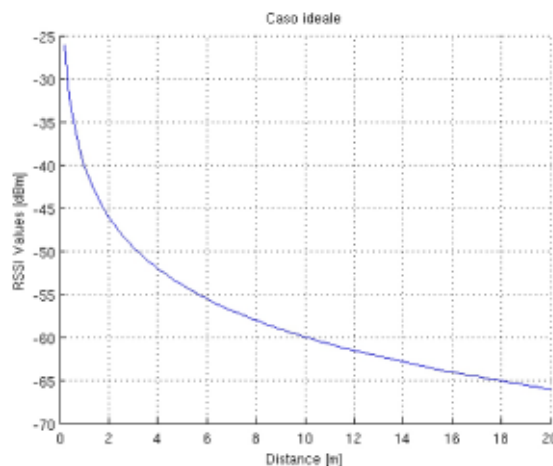


Figura 1.3: RSSI - Andamento della potenza in funzione della distanza percorsa dal segnale

La stima della potenza del segnale ricevuto è data dall'indicatore RSSI [9]. La distanza emettitore-ricevitore si stima utilizzando **l'equazione di trasmissione di Friis**.

$$P_T = P_R \frac{G_T G_R \lambda^2}{(4\pi)^2 d^n} \quad (1.1)$$

dove:

- P_R : potenza del segnale ricevuto (Watt)
- P_T : potenza del segnale trasmesso (Watt)
- G_R : guadagno dell'antenna ricevente
- G_T : guadagno dell'antenna trasmittente
- $\lambda = \frac{v}{f}$: lunghezza d'onda, dove v è la velocità di propagazione e f è la frequenza dell'onda
- d : distanza espressa in metri
- n : costante di propagazione del segnale che dipende dall'ambiente

Con la seguente equazione invece è possibile convertire la potenza espressa in Watt nella potenza espressa in dBm:

$$P[dBm] = 10 \log_{10}(10^3 P[W]) \quad (1.2)$$

Combinando l'equazione 1.1 con 1.2 e applicando le proprietà dei logaritmi si ottiene:

$$RSSI = -(10n \log_{10} d - A) \quad (1.3)$$

dove A è la potenza del segnale ricevuto a distanza fissa di un metro (espressa in dBm), considerando una costante di propagazione n .

La stima della distanza si ottiene infine dalla seguente equazione:

$$d = 10^{\left(\frac{A - RSSI}{10n}\right)} \quad (1.4)$$

Tuttavia la distanza restituita non è del tutto precisa, infatti la potenza del segnale potrebbe essere alterata dall'ambiente circostante attraverso i fenomeni di **Riflessione** (il segnale sbatte e si riflette su vari ostacoli seguendo più percorsi) e di **Assorbimento** (il decadimento viene alterato dagli oggetti presenti). Tale tecnica viene solitamente completata utilizzando il metodo della **Trilaterazione** (sez.1.4.2)

1.3.1.2 Time Of Arrival measurements

A differenza del precedente metodo, con questa tecnica la distanza tra emettitore e ricevitore viene stimata sulla base del tempo impiegato dal segnale a raggiungere il ricevitore. Nello specifico la sequenza di azioni è:

1. Il nodo *A* invia il segnale al tempo t_1
2. Il segnale arriva al nodo *B* al tempo t_2
3. *B* elabora il messaggio impiegando un tempo t_d e lo invia al tempo t_3
4. Il segnale torna al nodo *A* al tempo t_4

Come mostrato dalla figura seguente:

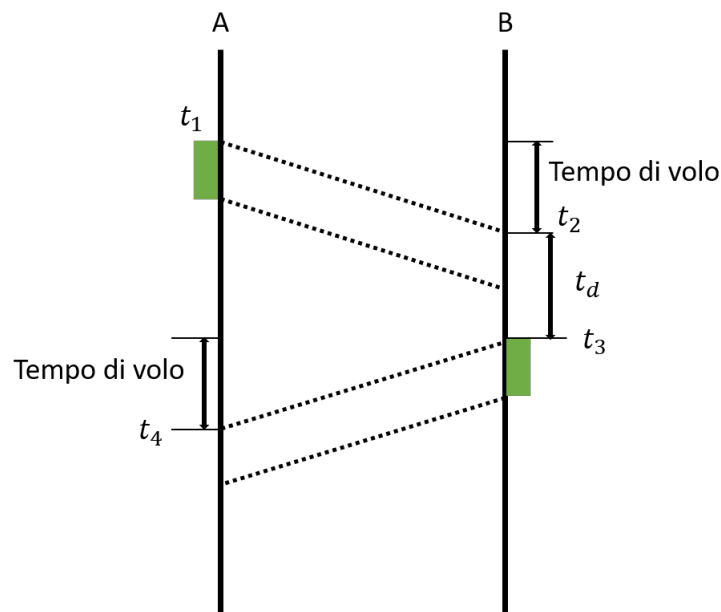


Figura 1.4: ToA - Principio di funzionamento

Quindi il tempo di viaggio può essere ricavato con la seguente equazione:

$$t_d = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \quad (1.5)$$

E infine la distanza stimata attraverso:

$$d_{ToA} = t_d * c \quad (1.6)$$

dove c è la velocità di propagazione della luce nel vuoto pari a 299792458 m/s. Per identificare in modo univoco un target, questa tecnica viene completata dalla tecnica di posizionamento nota come **Trilaterazione** (sez. 1.4.2), come per le misure RSSI viste precedentemente.

Il difetto principale di questa tecnica consiste nel fatto che sistemi utilizzati devono avere un complesso meccanismo di sincronizzazione per mantenere una fonte affidabile di tempo per i sensori[10].

1.3.1.3 Time Difference Of Arrival

Questa tecnica è basata sulla differenza nel tempo di arrivo di un segnale emesso da due sorgenti diverse verso un altro nodo, come mostrato in Fig.1.5).

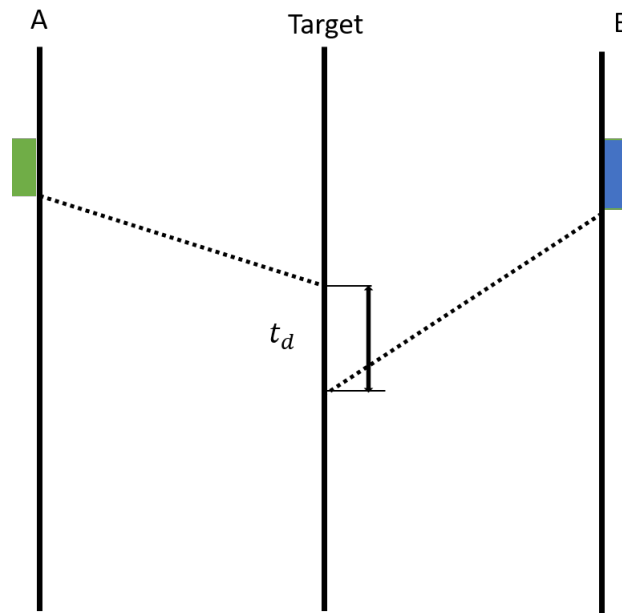


Figura 1.5: TDoA - Principio di funzionamento

Se si suppongono note le posizioni dei nodi A e B rispetto ad un sistema di riferimento, indicate rispettivamente dalle tuple (x_B, y_B) e (x_A, y_A) , la distanza del nodo *Target* può essere stimata dalla seguente equazione:

$$\Delta d = \Delta t_d * c \quad (1.7)$$

Dove:

- c è la velocità di propagazione della luce nel vuoto
- Δt è la differenza del tempo di arrivo dei segnali emessi dai nodi A e B
- Δd è la distanza in due dimensioni: $(\sqrt{(x_B - x)^2 + (y_B - y)^2} - \sqrt{(x_A - x)^2 + (y_A - y)^2})$

In questo modo, la posizione del *target* viene stimata all'interno del luogo geometrico dei punti del piano aventi come costante la differenza delle distanze tra i nodi, ovvero dall'iperbole avente come fuochi i nodi A e B . Come mostrato in Fig.

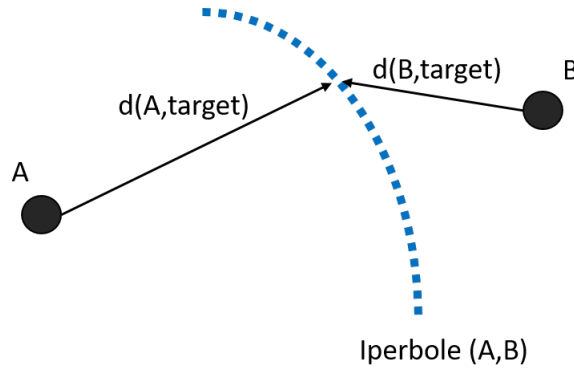


Figura 1.6: TDoA - Stima della posizione lungo l'iperbole identificata da due nodi

Tuttavia così facendo la posizione del *target* rimane stimata in un'insieme di punti infinito, quindi come per le tecniche viste precedentemente (1.3.1.1 e 1.3.1.2) per identificare in modo univoco il target questa tecnica ha bisogno di essere completata dalla tecnica di posizionamento nota come **Trilaterazione** 1.4.2.

1.3.2 Angle Based

1.3.2.1 Angle of Arrival

Con questa tecnica la posizione del *target* viene stimata misurando gli angoli di incidenza del segnale trasmesso ad altri nodi.

Si consideri [11] un antenna in un canale di propagazione, la tensione del segnale ricevuto (1.7) è data dalla seguente equazione:

$$V = \int_0^{2\pi} A_o A(\varphi) G(\varphi) d\varphi \quad (1.8)$$

Dove:

- $AoA(\varphi)$ rappresenta l'ampiezza e la fase dell'onda incidente
- $G(\varphi)$ è il campo elettrico del nodo target
- C valore proporzionale costante

Se si ruota l'antenna di un angolo α intorno a se stessa nel piano cartesiano la precedente equazione diventa:

$$V(\alpha) = \int_0^{2\pi} AoA(\alpha - \varphi)G(\varphi)d\varphi \quad (1.9)$$

Possiamo notare che l'Eq.1.9 è la convoluzione di AoA e G e può essere scritta nel seguente modo:

$$V(\alpha) = CAoA(\alpha) * G(\alpha) \quad (1.10)$$

Si è scelto di normalizzare l'Eq.1.10 con il valore costante C . Quindi, utilizzando la trasformata di Fourier l'Eq.1.10 diventa:

$$F(V(\alpha)) = F(AoA(\alpha))F(G(\alpha)) \quad (1.11)$$

Da cui è possibile calcolare l'angolo di incidenza desiderato:

$$AoA(\alpha) = F^{-1} \frac{F(V(\alpha))}{F(G(\alpha))} \quad se \quad F(G(\alpha)) \neq 0 \quad (1.12)$$

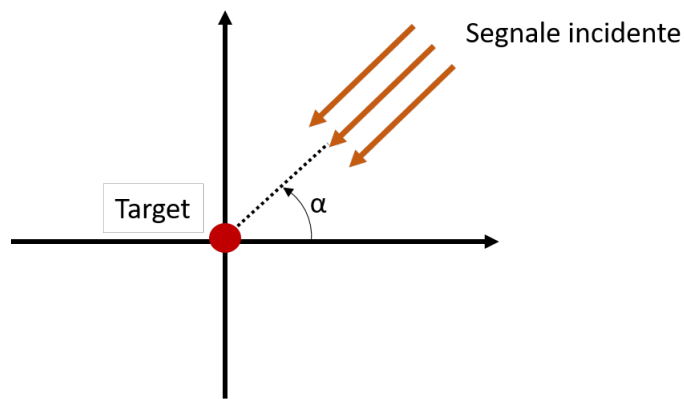


Figura 1.7: AoA - Stima della posizione attraverso l'angolo di incidenza

Il vantaggio dell'AoA risiede nella possibilità di ottenere un risultato attendibile senza la necessità di informazioni riguardanti i tempi di trasmissione. A fronte del

risparmio dal punto di vista computazionale, la tecnica presenta alcuni svantaggi pratici, dovuti al costo dell'hardware, che al fine di restituire informazioni precise, deve essere di alta qualità; rischiando altrimenti di incorrere in fenomeni che comprometterebbero la misurazione. Per questo motivo solitamente questa tecnica viene completata dalla tecnica di posizionamento nota come *Triangolazione* (1.4.3).

1.4 Tecniche di localizzazione

Per tecniche di localizzazione si intendono tutte quelle tecniche che, combinate alle differenti metodologie di stima della posizione viste precedentemente (vedi 1.3), permettono di localizzare un nodo *target* all'interno di un sistema di riferimento. In questo paragrafo vengono illustrate quelle più conosciute e basilari nell'ambito degli IPS.

1.4.1 MIN-MAX

Combina le stime della distanza di più *anchor*, ottenute attraverso tecnica *RSSI* (1.3.1.1), nel seguente modo:

- Stimare la distanza d_i di ogni nodo i -esimo in base al valore *RSSI*
- Traccia due linee orizzontali e verticali a distanza d_i dallo nodo *target*
- Identifica un quadrato di lato $2 d_i$ i cui estremi saranno:
 $[max(x_i - d_i), max(y_i - d_i)] * [min(x_i + d_i), min(y_i + d_i)]$
- Calcola le intersezioni dei quadrati

Il centro del quadrato (Fig.1.8) rappresenta la posizione stimata del *target*. Più piccola sarà l'area e maggiore sarà l'accuratezza della posizione stimata.

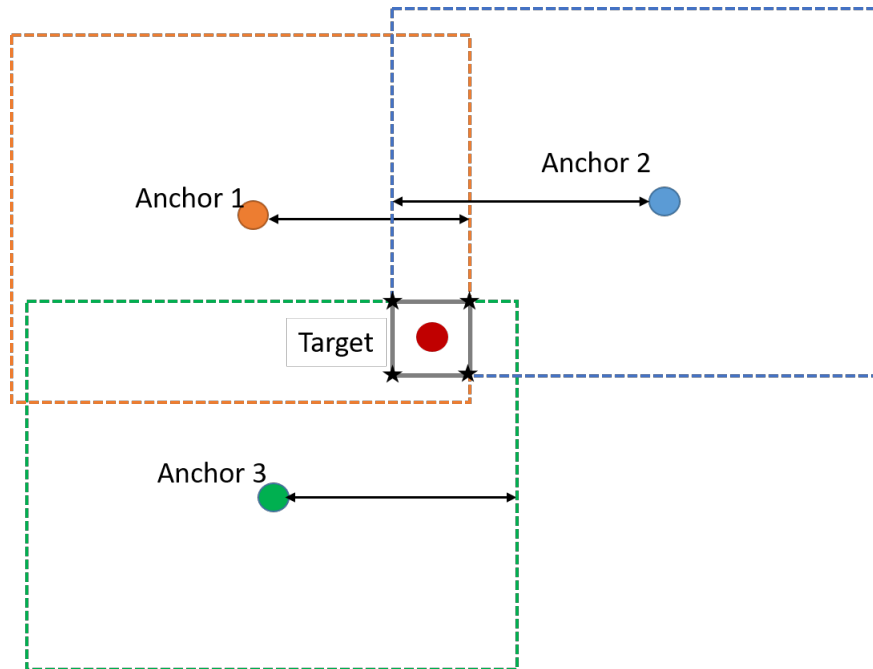


Figura 1.8: MIN-MAX - Tecnica di posizionamento

1.4.2 Trilaterazione

Consideriamo 3 *Anchor* intorno cui disegniamo 3 circonferenze aventi per centro le coordinate degli *Anchor* e per raggio l'RSSI del segnale ricevuto dallo *Unknown*, come mostrato in figura:

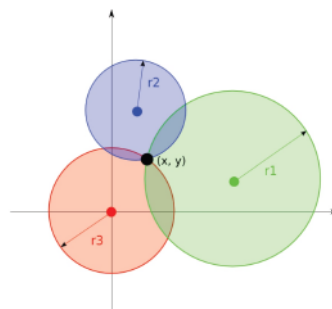


Figura 1.9: Trilaterazione - Esempio esplicativo

Quindi le coordinate dell'Unknown sono la soluzione del seguente sistema:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = r_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = r_3^2 \end{cases} \quad (1.13)$$

In base alla soluzione del sistema si può avere una delle seguenti situazioni:

- La soluzione non è unica, si hanno tre cerchi che si sovrappongono
- Il sistema non ammette soluzione, i raggi vanno aumentati
- La soluzione esiste ed è unica, i tre cerchi si intersecano in un solo punto

1.4.3 Triangolazione

A differenza delle Trilaterazione (1.4.2), queste tecniche identificano la posizione del nodo *target* a partire dagli angoli stimati da tre anchors attraverso una delle tecniche Angle Based (1.3.2).

In [12] si descrive la triangolazione geometrica attraverso il seguente algoritmo:

1. Siano $1, 2$ e 3 le anchor in grado di stimare l'angolo, rispetto ad una circonferenza concentrica all'anchor stessa, del nodo *target*
2. siano L_{12} e L_{31} rispettivamente le distanze tra l'anchor 1 e 2 e l'anchor 3 e 1
3. Siano gli angoli compresi tra 1 e 2 e tra 1 e 3 , indicati rispettivamente con λ_{12} e λ_{13} , minori di 180°
4. sia ϕ l'angolo tra l'asse x positivo e la linea formata dall'anchor 1 e 2
5. sia σ l'angolo tra l'asse x positivo, l'anchor 1 e l'anchor 3 più ϕ
6. sia $\gamma = \sigma - \lambda_{31}$
7. sia $p = \frac{L_{31} \sin \lambda_{12}}{L_{12} \sin \lambda_{31}}$
8. sia $\tau = \tan^{-1} \frac{\sin \lambda_{12} - p \sin \gamma}{p \cos \gamma - \cos \lambda_{12}}$

9. sia $L_1 = \frac{L_{12} \sin(\tau + \lambda_{12})}{\sin \lambda_{12}}$

Allora le coordinate x e y del target sono date da:

- $x_R = x_1 - L_1 \cos(\phi + \tau)$
- $y_R = y_1 - L_1 \sin(\phi + \tau)$
- $\Phi_R = \phi + \tau - \lambda_1$

Dove Φ_R rappresenta l'orientamento del nodo target, come mostrato in Fig.1.10.

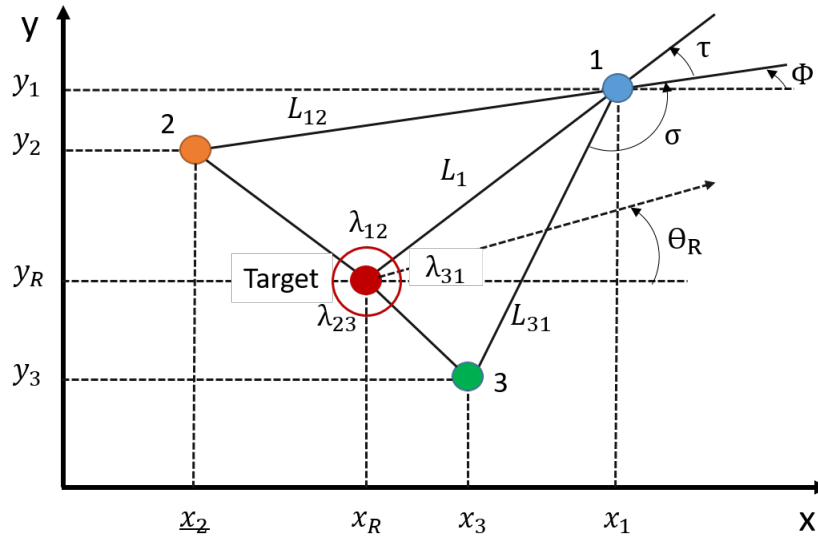


Figura 1.10: Triangolazione - Esempio esplicativo

1.5 Sensor Data Fusion

In generale con Sensor Data Fusion si indicano tutte quelle tecniche che [13] combinano dati provenienti da sensori o derivate da altre risorse tali che l'informazione risultante ha una minore incertezza rispetto a quella ottenuta utilizzando le risorse individualmente.

Nel contesto degli IPS (1.2), il Sensor Fusion ha come scopo quello di determinare informazioni riguardanti la posizione di oggetti e/o persone. Le risorse delle informazioni grezze sono le più svariate, tra le quali:

- Accelerometro
- Giroscopio
- Magnetometro
- Infrarossi
- RFID
- Sensori ottici
- Sensori di pressione
- Bluetooth
- WiFi
- Telecamera
- UWB

Come e quali risorse utilizzare per stimare la posizione di un oggetto è una sfida ingegneristica non banale, ma grazie alla crescente richiesta di IPSs la ricerca di nuove tecnologie e algoritmi in questo settore ha permesso di raggiungere risultati incredibili fino a qualche anno.

Capitolo 2

Descrizione del lavoro

Per comprendere la soluzione proposta e il sistema ideato si utilizzerà un approccio top-down. Partendo da un livello d'astrazione tale in cui emergono solo i requisiti funzionali e il sistema è rappresentato da una *black box*, si raggiunge un livello che permetterà di affinare i requisiti iniziali e mostrare i *sottosistemi* che lo costituiscono. Infine nell'ultimo livello avremo una visione dei *moduli* che compongono i sottosistemi e che sono stati realizzati nel contesto di questa tesi.

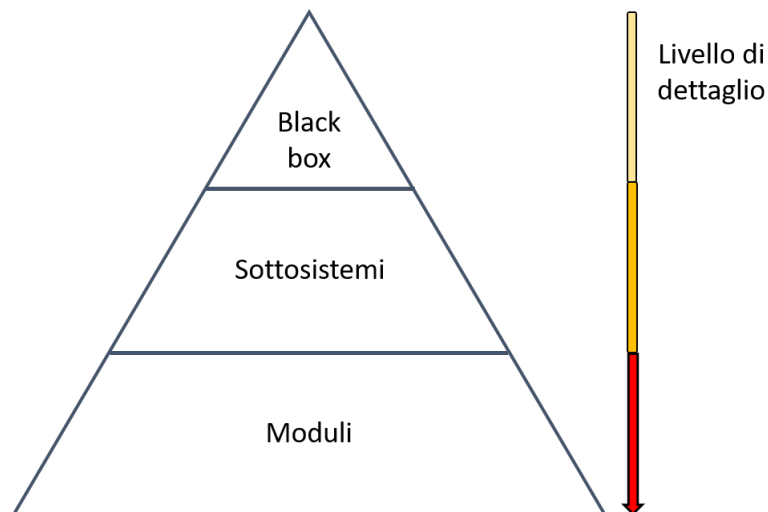


Figura 2.1: Rappresentazione dei livelli d'astrazione utilizzati per descrivere il sistema

2.1 Livello black box

Considerando il sistema in questione come una black box (Fig.2.2) e l'infrastruttura di rete in modo astratto, i due macro-requisiti funzionali sono rispettivamente:

- **R1:** Geolocalizzare l'operatore
- **R2:** Trasmettere messaggi predefiniti come stato della vittima e codici d'emergenza.

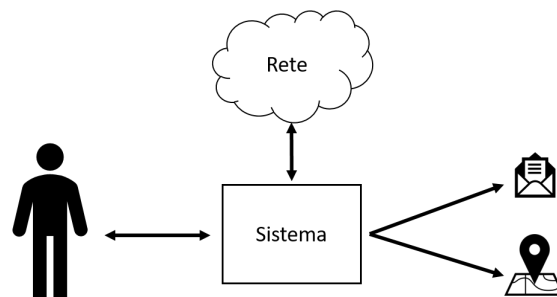


Figura 2.2: Rappresentazione del sistema come black box

Nelle fasi primordiali del progetto, si è scelto di allocare la maggior parte delle risorse lavorative nel completamento di **R1** lasciando ad una fase successiva lo sviluppo di **R2**. *R1* può essere suddiviso in due requisiti più specifici:

- **R1.1:** Determinare la posizione di un operatore all'interno della rete
- **R1.2:** Identificare il cammino minimo da un operatore ad un altro

Quest'ultimo rappresenta sia la possibilità da parte dell'operatore di eseguire il percorso all'inverso, sia la possibilità che venga raggiunto da una squadra di supporto. A questo punto si può scendere al livello d'astrazione successivo (2.1).

2.2 Livello sottosistemi

Per soddisfare il requisito **R1.1** è necessario che l'operatore si trovi all'interno di una zona *infrastrutturata*, ovvero una zona dove i nodi della rete siano georeferenziati rispetto ad un sistema di riferimento comune. Considerata la modalità con la quale la rete è costruita, ovvero dinamicamente da un'esploratore, garantire tale situazione non è un compito banale.

La soluzione si costruisce per iterazione georeferenziando i singoli nodi nel momento in cui vengono aggiunti dall'esploratore.

Con riferimento all'esempio illustrato precedentemente (Fig.3- Fig.8), si consideri lo step 2. Per ipotesi si supponga che il primo nodo sia già georeferenziato, nel momento in cui l'operatore risulti essere al limite della line-of-sight e/o della distanza di sicurezza piazzerà il secondo nodo.

Per mantenere il livello d'astrazione attuale ci basta sapere che le caratteristiche della tecnologia utilizzata nell'implementazione della rete, fa sì che le singole celle abbiano un raggio d'azione all'interno del quale il sottosistema *SI* (Fig.2.4) può calcolare la distanza tra l'operatore e il centro della cella di appartenenza.

Tale informazione non è però sufficiente, infatti ci sono infiniti punti sulla circonferenza con centro nel primo nodo e raggio pari alla distanza. Per poter georeferenziare in modo univoco il secondo nodo si deve trovare anche l'angolo in riferimento al primo nodo.

La figura seguente rappresenta un tentativo di georeferenziare il secondo nodo utilizzando soltanto la distanza tra i due nodi.

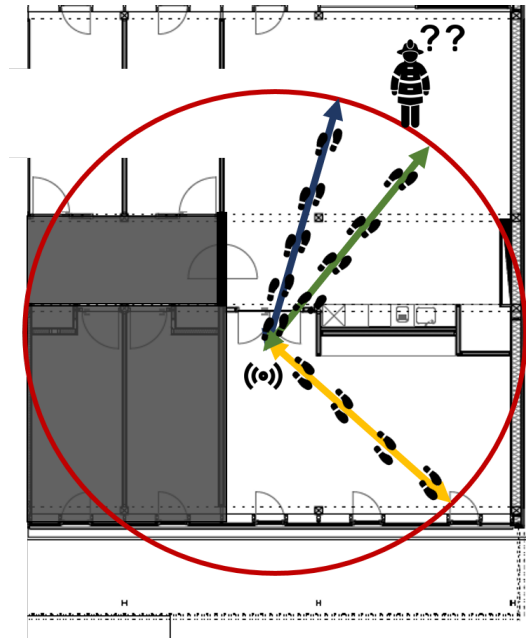


Figura 2.3: Ambiguità nella georeferenziazione di un secondo nodo utilizzando soltanto la distanza dal precedente

Nell'esempio appena proposto, si sono mostrate solo tre delle possibili infinite posizioni del secondo nodo. Si assuma che il punto corretto sia l'intersezione tra il vettore in blu e la circonferenza. In tal caso un'ambiguità con il vettore in verde potrebbe essere accettata in quanto si discosta di pochi metri dalla reale posizione, ben diversa sarebbe un'ambiguità con il vettore in giallo che renderebbe l'informazione del tutto errata e il sistema disinformatore.

In conclusione per georeferenziare in maniera univoca il nuovo nodo (rispetto al primo) sono necessarie le seguenti informazioni:

- La distanza tra i due nodi
- L'angolo tra i due nodi

Per ricavare queste informazioni si sono ideati due sottosistemi, rappresentati dalla Fig.2.4:

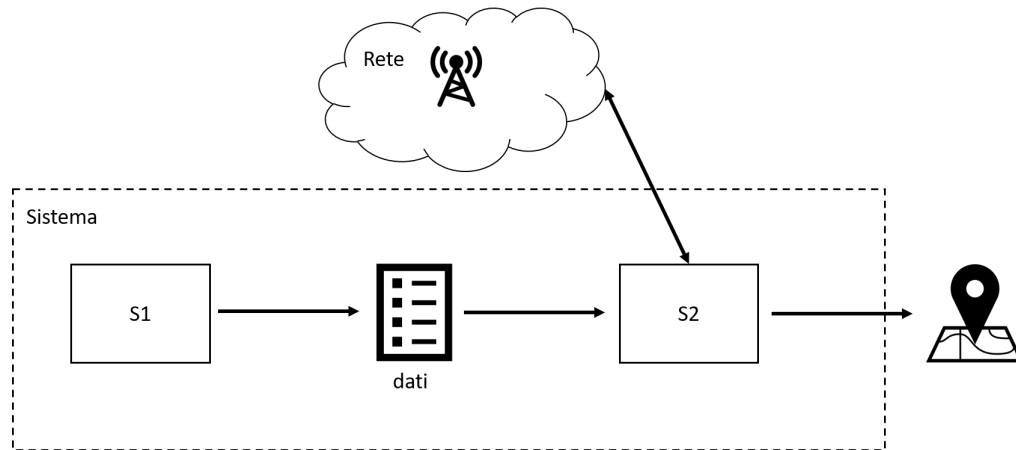


Figura 2.4: Rappresentazione dei sottosistemi necessari per georeferenziare i nodi della rete

Ognuno dei quali con la propria responsabilità:

- **S1**: ha il compito di ricavare tramite i sensori (cap.3) le informazioni riguardanti gli angoli assunti dall'esploratore lungo il tragitto
- **S2**: ha il compito di ricevere i dati da S1 e la distanza dal nodo precedente, elaborarli e infine determinare la posizione del nuovo nodo

Relativamente alla rappresentazione del sistema mostrata in Fig.2.4, il lavoro di questi tesi si colloca nella progettazione e nello sviluppo del sottosistema S1.

Nel prossimo paragrafo si raggiungerà il livello d'astrazione più basso della piramide (vedi 2.1) dettagliando i moduli che compongono il sottosistema realizzato.

2.3 Livello moduli

Prima di "aprire" il sottosistema è bene specificare che si utilizzerà il termine *modulo* per riferirsi sia a componenti hardware che software. Questo abuso di notazione permetterà di mostrare con un unico livello d'astrazione tutti i moduli progettati e realizzati al fine di stimare gli angoli assunti dall'esploratore lungo il tragitto (si veda 2.2).

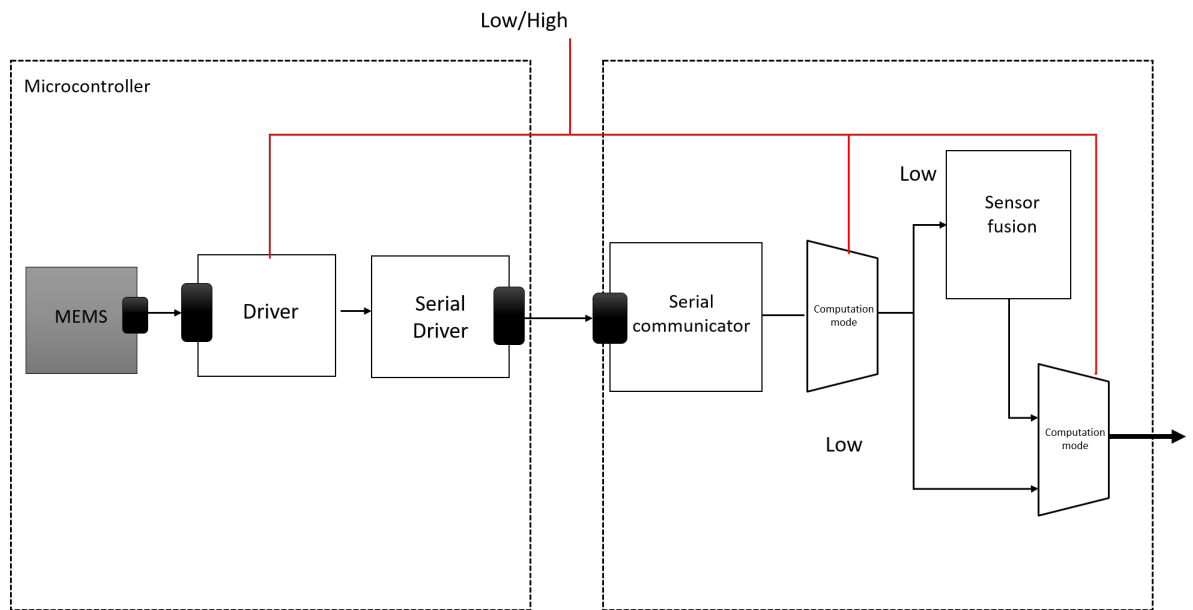


Figura 2.5: Rappresentazione dei sottosistemi in moduli

Per esplicitare al meglio i compiti dei singoli moduli e mantenere l'attuale livello d'astrazione, fermo restando che tutti i dettagli tecnici verranno forniti nei capitoli successivi, si mostrerà il flusso di dati generato in uno qualsiasi degli intervalli di campionamento lungo il tragitto dell'operatore tra un nodo e l'altro.

Il flusso inizia nel momento in cui il modulo *driver* acquisisce le informazioni dall'unità *MEMS* (cap.3) riguardanti la velocità angolare, l'accelerazione e il campo magnetico relativi all'operatore. Come mostrato in Fig.2.6.

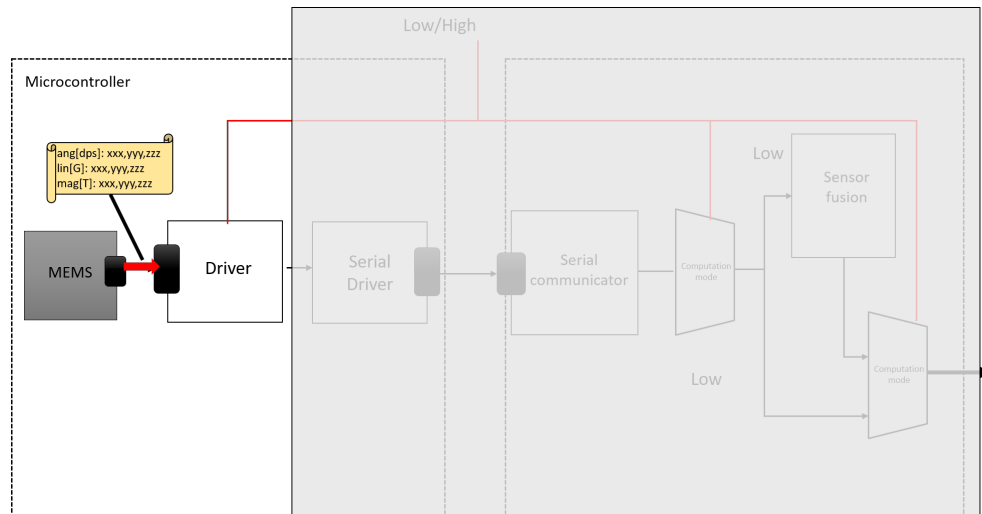


Figura 2.6: Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 1

I dati acquisiti sono "grezzi" e devono essere elaborati (Cap.4). Quando e da chi questa computazione verrà eseguita durante il flusso, viene stabilito attraverso il comando Low/High. Da questo ne consegue anche la modalità di funzionamento del driver in:

- **HCM:** High Computation mode, i dati vengono elaborati dal microcontrollore
- **LCM:** Low Computation mode, i dati verranno elaborati in seguito dal sottosistema *App*

La scelta tra quale di queste due modalità utilizzare verrà motivata nel capitolo riguardante l'analisi dei risultati (cap.4), per il momento si ipotizzi di settare la linea di comando al valore "Low" e quindi di utilizzare il driver in *LCM*. Con queste impostazioni i dati grezzi vengono impacchettati ed etichettati con un timestamp relativo, prima di essere inviati dal modulo *Serial driver* e ricevuti dal modulo *App* mediante il modulo *Serial communicator*, quest'ultimo li inoltra all'ingresso del multiplexer *Computation mode* come mostrato in Fig.2.7:

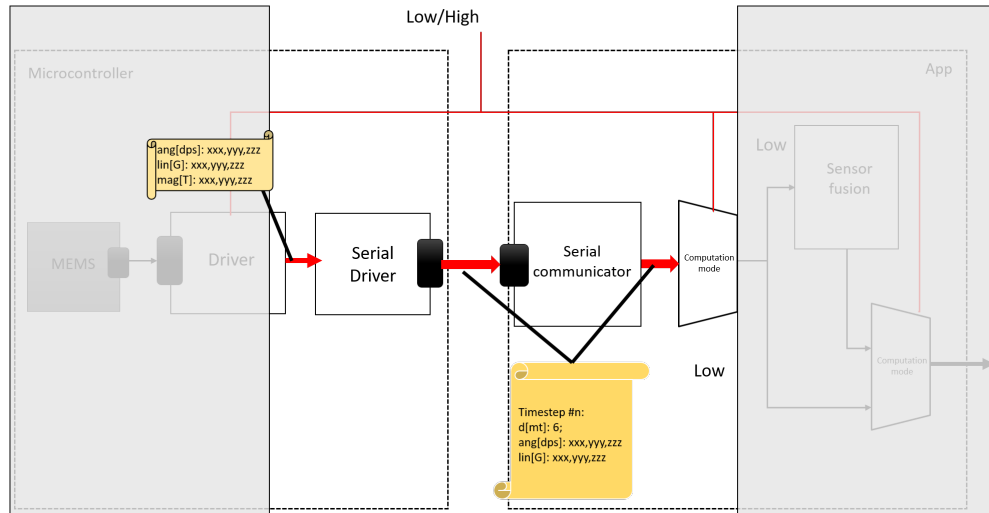


Figura 2.7: Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 2

Poiché per ipotesi si è scelto di settare la linea di comando sul valore **Low**, il multiplexer devia il flusso di dati verso il modulo *Sensor Fusion* come mostrato in figura:

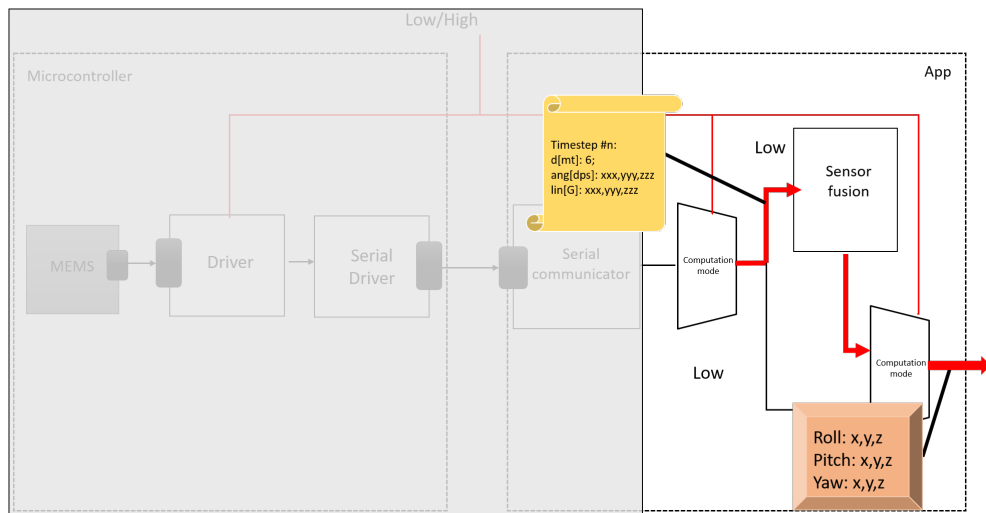


Figura 2.8: Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 3

A questo punto il modulo *Sensor Fusion* elabora i dati grezzi (vedi cap.4.2) fornendo in uscita gli angoli di Eulero (Cap.4.1) relativi all'operatore:

- **Roll**
- **Pitch**
- **Yaw**

Questi dati verranno usati come input dal sottosistema S2 (si veda 2.4) che provvederà, insieme all'informazione relativa alla distanza dal nodo precedente, a determinare l'angolo del nuovo nodo e quindi a risolvere il problema di georeferenziazione emerso nel livello di astrazione precedente (si veda 2.2).

Capitolo 3

Unità di misura inerziale

Le unità di misura inerziale [14] (in inglese Inertial Measurement Units - IMU) sono dispositivi elettronici basati su sensori inerziali come accelerometri (3.1) e giroscopi (3.2). In molti casi a questi vengono aggiunti altri sensori utili ad applicazioni di navigazione come il magnetometro (3.3). Nello specifico di questa tesi, l'IMU utilizzata è un circuito integrato composto da questi tre sensori (più altri non utilizzati come sensore di temperatura) realizzati tramite tecnologia MEMS (acronimo di Microelectro Mechanical System, ovvero sistemi meccanici microelettrici).

Nel corso degli anni l'interesse per questa tecnologia è cresciuto grazie ai vantaggi in termini economici e tecnici, tra questi i più importanti sono:

- costo di realizzazione costante e proporzionale alla superficie del dispositivo
- grande potenziale di integrazione nei circuiti elettronici integrati
- basso consumo energetico
- dimensioni ridotte

In questo capitolo si illustrano i principi di funzionamento alla base dei sensori, realizzati mediante tecnologia MEMS, integrati nell'IMU utilizzata nel lavoro di questa tesi.

3.1 Accelerometro

In generale un accelerometro è un dispositivo in grado di misurare l'accelerazione di un corpo rigido causata da una forza esterna. Questa può essere statica, come la

forza di gravità, o dinamica nel caso di forze vibranti applicate al dispositivo. Uno dei più comuni accelerometri MEMS è quello *capacitivo* che, come il nome suggerisce, si basa sulla *capacità* elettrostatica. Se due piastre sono posizionate parallelamente tra di loro e poste ad una certa distanza, allora la capacità generata è data da:

$$C = \varepsilon_r \varepsilon_0 \frac{A}{d} \quad (3.1)$$

Dove:

- C è la capacità
- ε_0 è la costante dielettrica del vuoto
- ε_r è la costante dielettrica relativa al materiale utilizzato per le piastre
- A è l'area delle piastre
- d è la distanza tra le due piastre

Dall'Eq.3.1 si noti che la capacità può variare solo se vi sono cambiamenti nell'area delle piastre o nella loro distanza. Proprio su quest'ultimo parametro si basano gli accelerometri capacitivi. Una classica struttura è rappresentata dalla Fig.3.1:

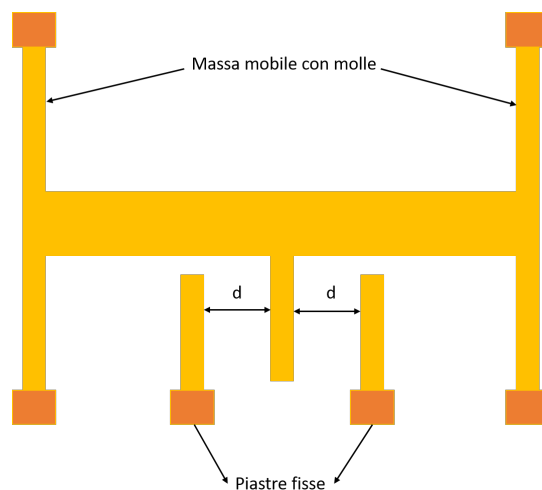


Figura 3.1: Rappresentazione esemplificativa di un accelerometro capacitivo a riposo

La massa centrale è in grado di muoversi lungo un asse orizzontale grazie a delle molle poste alle sue estremità, come rappresentato in Fig.3.2:

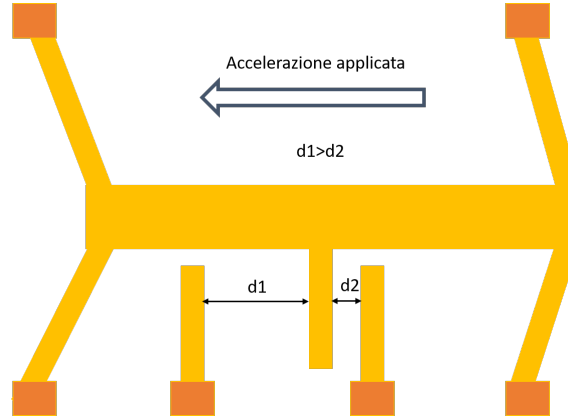


Figura 3.2: Rappresentazione esemplificativa di un accelerometro capacitivo che subisce una forza esterna

A seguito del movimento della massa centrale, la distanza $d2$ in Fig.3.2 si farà più piccola provocando una variazione di tensione ai capi della massa centrale. Questa verrà quindi convertita in un valore numerico rappresentante l'accelerazione subita in base alla scala e alla sensibilità del dispositivo.

3.2 Giroscopio

In generale, un giroscopio è un dispositivo in grado di misurare la velocità angolare a cui è sottoposto il dispositivo. I giroscopi realizzati mediante tecnologia MEMS si basano sulla forza di *Coriolis*. In fisica [18], la forza di Coriolis è una forza apparente a cui risulta soggetto un corpo quando si osserva il suo moto da un sistema di riferimento che sia in moto circolare rispetto ad un sistema di riferimento inerziale. I giroscopi di questo tipo sono composti [19] da una *massa* m , due *molle* e due ammortizzatori come mostrato in Fig.3.3. Si assuma l'asse x come l'asse di direzione (drive mode) e l'asse y come l'asse di rilevamento (sensing mode). Quando la massa è sottoposta ad una vibrazione armonica applicata da una forza elettrostatica, elettromagnetica o elettrotermica, lo spostamento lungo l'asse x è dato da:

$$x(t) = A_x \cos(\omega_x t) \quad (3.2)$$

Dove A_x è l'ampiezza e ω_x è la frequenza angolare. Una velocità angolare Ω_z in input intorno all'asse z causa un'accelerazione di Coriolis lungo l'asse y data dalla

seguente equazione:

$$a_y = 2\Omega_z \times \frac{dx}{dt} = -2\Omega_z A_x \omega_x \sin(\omega_x t) \quad (3.3)$$

La massa quindi inizierà a vibrare lungo l'asse y a causa della forza di Coriolis e la velocità angolare Ω_z può essere calcolata misurando lo spostamento lungo l'asse vibrante.

Quando il *drive mode* e il *sense mode* sono perfettamente uguali ($\omega_x = \omega_y$), l'ampiezza lungo l'asse y raggiunge il massimo mentre la larghezza di banda raggiunge il minimo. In generale, questi due parametri dovrebbero essere uguali al fine di ottimizzare la sensibilità e la larghezza di banda.

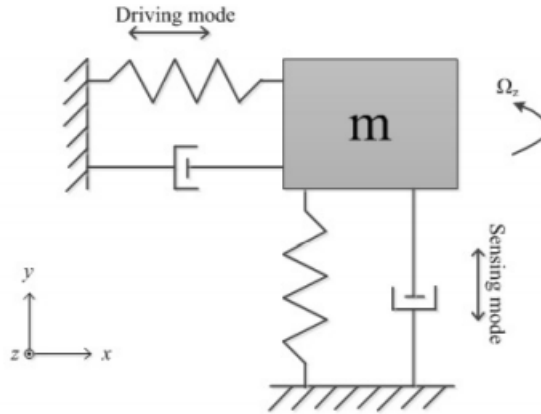


Figura 3.3: Rappresentazione di un giroscopio vibrante per la misura della velocità angolare lungo l'asse z

3.3 Magnetometro

In generale, un magnetometro è un dispositivo in grado di rilevare l'intensità e la direzione del campo magnetico presente. Questi si basano sulla ben nota forza di Lorentz.

Se si eroga una certa corrente in un conduttore posto in un campo magnetico trasversale al flusso, si genera una forza proporzionale alla velocità dei portatori, alla carica ed al valore del campo magnetico, diretta nella direzione ortogonale ad entrambi, secondo la relazione:

$$\vec{F}_L = q \vec{v} \times \vec{B} \quad (3.4)$$

Dove:

- q è la carica elementare
- F è la forza di Lorentz
- B è il campo magnetico nel vuoto

Indicando con l la lunghezza del conduttore si ha:

$$\vec{F}_L = l \vec{i} \times \vec{B} \quad (3.5)$$

Questa forza viene quindi sfruttata per misurare il campo magnetico esterno agente sul dispositivo. Una delle realizzazioni più comuni nell'ambito dei sensori realizzati mediante tecnologia MEMS sono i magnetometri capacitivi.

In Fig.3.4 è mostrata una semplice implementazione di un magnetometro capacitivo. Questo è composto da due *molle* i cui terminali sono ancorati al substrato del dispositivo dove viene fatta circolare una corrente elettrica. Nel punto centrale le molle mantengono sospeso un *rotore* dove idealmente non scorre corrente e al cui interno sono ancorati degli *statori*. In presenza di una forza di Lorentz, le *molle* si deformano dando origine ad uno spostamento rigido del *rotore*. Poiché lo *statore* non si muove, la capacità tra il *rotore* e lo *statore* cambia in maniera proporzionale all'intensità del campo magnetico esterno.

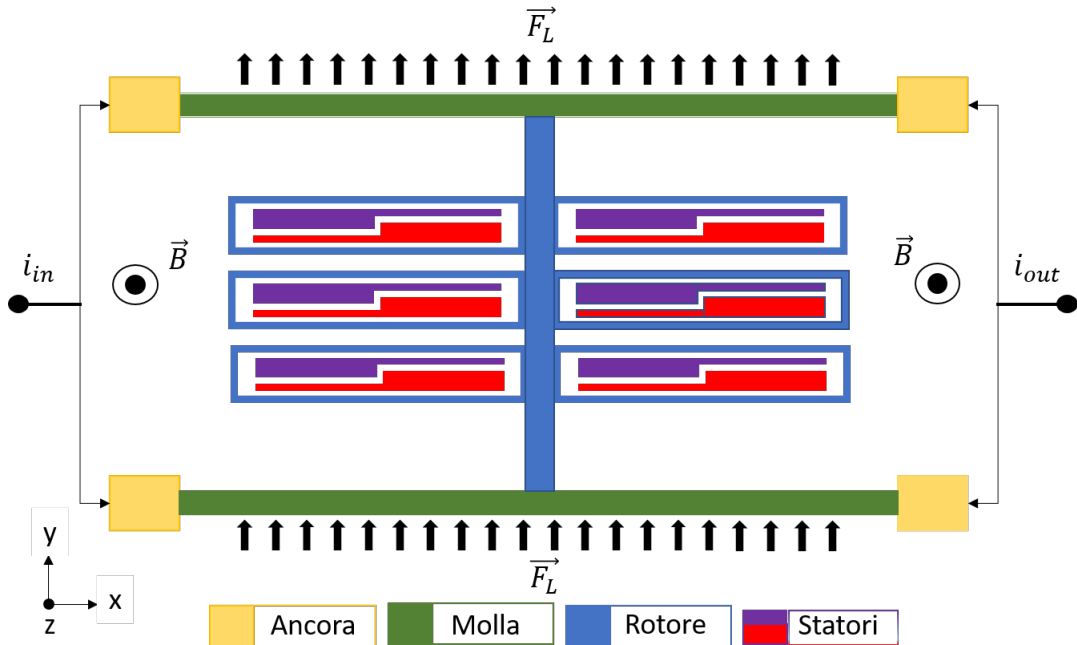


Figura 3.4: Rappresentazione esemplificativa di un magnetometro capacitivo lungo l'asse z

3.4 Modello di misura

Dopo aver introdotto i principi di funzionamento dei sensori è bene specificare il modello di misurazione.

I sensori utilizzati hanno tre assi lungo i quali una "quantità fisica" (esempio forza, velocità angolare, campo magnetico) è convertita in un segnale di tensione in uscita. Tipicamente questi sensori hanno un comportamento lineare nell'area di lavoro. Sulla base di questa osservazione, la seguente equazione (semplificata) descrive la relazione tra la forza fisica $y(t)$ e la tensione in uscita dal sensore $u(t)$:

$$u(t) = GRy(t) + c \quad (3.6)$$

Dove:

- G è la matrice diagonale contenente il guadagno per ogni asse sensibile
- R è la matrice di allineamento che specifica la direzione degli assi
- c è il vettore di offset

Al fine di discutere il modello di misura, si devono introdurre i seguenti sistemi di coordinate (in inglese: coordinate frames) rappresentati in Fig.3.4:

- Il **frame del corpo** - *b-frame* (in inglese: *body frame*): è il sistema di riferimento dei movimenti dell'IMU. L'origine è posta al centro dei sensori e allineata al case posto sul chip. Tutte le misure inerziali sono calcolate su questo sistema di riferimento.
- Il **frame di navigazione** - *n-frame* (in inglese: *navigation frame*) è il frame geografico nel quale vogliamo navigare. Per navigazioni a corto raggio è considerato statico rispetto alla terra.
- Il **frame inerziale** - *i-frame* (in inglese: *inertial frame*): è un frame stazionario non rotante. L'IMU misura le forze relativamente a questo frame. La sua origine è posta al centro della terra e i suoi assi sono allineati rispetto alle stelle.
- Il **frame terrestre** - *e-frame* (in inglese: *earth frame*): coincide con l' *i-frame* ma ruota intorno alla terra. L'origine è posta al centro della terra e gli assi fissati rispetto ad essa.

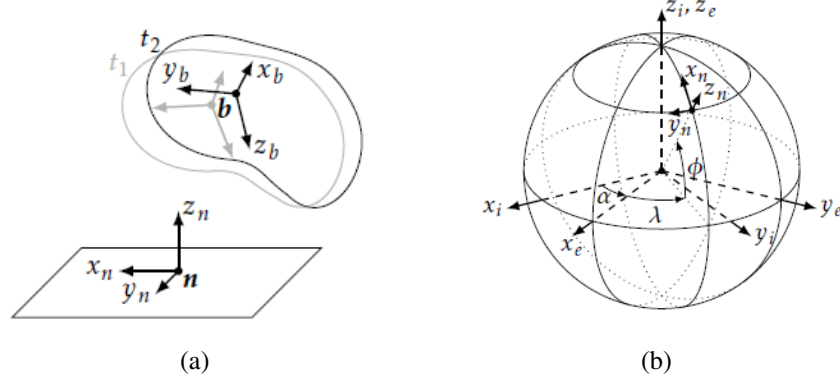


Figura 3.5: in 3.5(a) il *b-frame* nell'istante t_1 e t_2 relativamente al *n-frame*, in 3.5(b) l'*n-frame* in latitudine φ e longitudine λ , l'*e-frame* all'angolo $\alpha(t) = \omega_{ie}t$ e l'*i-frame*

Ignorando la dipendenza dal tempo delle quantità coinvolte, la misura del giroscopio (si veda 3.2) è modellata in [17] come:

$$y_\omega = \omega_{ib}^b + \delta_\omega^b + e_\omega^b \quad (3.7)$$

Dove:

- ω_{ib} è la velocità angolare nel *b-frame* osservata dall'*i-frame*
- δ_ω è la deriva del sensore che varia lentamente nel tempo
- e_w^b è il rumore gaussiano

La velocità angolare ω_{ib} può essere così estesa:

$$\omega_{ib} = R^{bn}(\omega_{ie}^n + \omega_{en}^n) + \omega_{nb}^b \quad (3.8)$$

Dove:

- R è la matrice di rotazione
- ω_{ie} è la velocità angolare della terra
- ω_{en} è velocità angolare di trasporto
- ω_{nb} è la velocità angolare richiesta ai fini della navigazione

La misura dell'accelerometro y_a è invece modellata in [17] come:

$$y_a = f^b + \delta_a^b + e_a^b = R^{bn}(\ddot{b}_{ii}^n - g^n) + \delta_a^b + e_a^b \quad (3.9)$$

Dove:

- f è la specifica forza esterna
- δ_a è la deriva del sensore che varia lentamente nel tempo
- e_a è il rumore gaussiano

L'Eq.3.9 divide la forza specifica nei suoi contributi provenienti dall'accelerazione lineare del corpo osservata dall'*i-frame* (\ddot{b}_{ii}) e dal vettore gravitazione g . L'accelerazione lineare può a sua volta essere espansa come:

$$\ddot{b}_{ii} = \omega_{ie}^n \times \omega_{ie}^n \times R^{ni} b^i + 2\omega_{ie}^n \times \dot{b}_n^n + \ddot{b}_{nn}^n \quad (3.10)$$

dove \ddot{b}_{nn} è l'accelerazione del corpo osservata dal *n-frame* richiesto per la navigazione.

Infine per il magnetometro (si veda 3.3) la misura y_m è così modellata:

$$y_m = m^b + e_m^b = R^{bn} m^n + e_m^b \quad (3.11)$$

Dove:

- m è il vettore del campo magnetico locale
- e_m è il rumore gaussiano

In assenza di oggetti ferromagnetici, m è il campo magnetico della terra e la misura del magnetometro può essere usata come una bussola per trovare la direzione del nord magnetico.

Capitolo 4

Stima dell'assetto tramite sensor fusion

In questo capitolo vengono inizialmente illustrati gli strumenti matematici utilizzati per rappresentare l'assetto di un corpo rigido nello spazio, successivamente viene illustrato l'algoritmo di fusione dei dati, provenienti dall'unità di misura inerziale, utilizzato per la stima dell'assetto dell'operatore.

4.1 Rappresentazione geometrica dell'assetto di un corpo rigido nello spazio

Con "*assetto di un corpo rigido*" si intende l'orientamento di un corpo rigido rispetto ad un particolare sistema di riferimento. Tale condizione può essere rappresentata attraverso numerosi strumenti matematici, tra i più utilizzati si hanno:

- Angoli di Eulero
- Quaternioni unitari

Nella Tab.4.1 vengono riportate sinteticamente le caratteristiche delle rappresentazioni appena enunciate [20]:

Rappresentazione	#Parametri	Caratteristiche
Angoli di Eulero	3	<ul style="list-style-type: none">- facilmente interpretabili dall'essere umano- funzioni trigonometriche nelle relazioni cinematiche- presentato singolarità in alcune situazioni- meno accurati dei quaternioni
Quaternioni unitari	4	<ul style="list-style-type: none">- non interpretabili facilmente dall'essere umano- equazioni della cinematica lineari- costo computazione di elaborazione minore degli angoli di Eulero- necessitano di un vincolo di norma unitaria

Tabella 4.1: Tabella comparativa delle rappresentazioni d'assetto

4.1.1 Angoli di Eulero

4.1.2 Quaternioni

4.2 Sensor fusion mediante un filtro di Kalman

Capitolo 5

Implementazione

5.1 Modalità di computazione

5.1.1 Low Computation mode

5.1.2 High Computation mode

5.1.3 Testing Computation mode

5.2 Canali di comunicazione

5.2.1 I2C

5.2.2 USB CDC

Capitolo 6

Analisi e validazione dei risultati

6.1 Analisi temporale

Questo tipo di analisi è stata effettuata al fine di determinare la frequenza massima entro la quale il microcontrollore è in grado di eseguire tutte le operazioni basilari, rappresentate in Fig.6.1. Ovvero per rispondere alla seguente domanda:

Qual'è la finestra temporale minima sufficiente a garantire che il microcontrollore sia in grado di eseguire correttamente le funzioni basilari?

Dove per funzioni basilari si intendono le operazioni di:

- **Lettura** dei dati grezzi dall'IMU tramite, indicato con T_{RX_I2C}
- **Elaborazione** dei dati grezzi (solo per le modalità *HCM* e *TCM*), indicato con $T_{MotionFX}$
- **Trasmissione** dei dati verso il modulo *App* (si veda 2.3), indicato con T_{TX_USB}

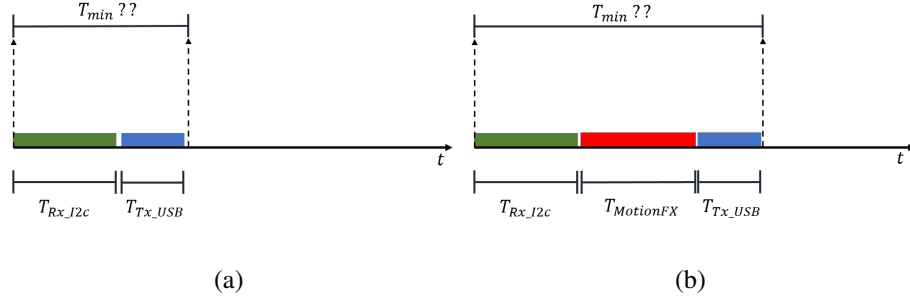


Figura 6.1: In 6.1(a) le operazioni basilari per la modalità **LCM**, in 6.1(b) le operazioni basilari per le modalità **HCM** e **TCM**

Per far ciò è opportuno analizzare singolarmente i due tempi, considerando le relative informazioni e la velocità del mezzo di comunicazione.

6.1.1 Analisi del tempo di lettura dei dati grezzi dall'IMU

Le informazioni riguardanti le grandezze fisiche misurate dai sensori integrati nell'IMU (Cap.3), tramite I2C (Cap.5.2.1), su un singolo asse del *b-frame* (Cap.3.4), sono rappresentate attraverso **16 bit**.

Di conseguenza il tempo di lettura T_{RX_I2C} è dato dalla seguente equazione:

$$T_{RX_I2C} = N_{sensori} \cdot N_{assi} \cdot T_{i_asse} \quad (6.1)$$

Dove:

- $N_{sensori}$ sono il numero di sensori utilizzati
- N_{assi} sono il numero di assi utilizzati dai sensori
- T_{i_asse} è il tempo di lettura del singolo asse

Dunque il problema si riduce alla determinazione di T_{i_asse} . Per come è implementata l'IMU, la lettura di un singolo asse si completa leggendo due registri da **8 bit**.

$$T_{i_asse} = 2 \cdot T_{RX_8bit} \quad (6.2)$$

Per stimare T_{RX_8bit} si è programmato il microcontrollore in modo tale da leggere 1000 volte un registro da 8 bit e misurare il tempo trascorso leggendo i *tick* di sistema. Per il codice si rimanda alla lettura dell'App.C.1.1.

Eseguendo il test svariate volte, si è osservato che il tempo necessario per leggere 1000 volte un registro da 8 bit è pari a $101ms$. Questo implica un *goodput* di $80Kb/s$ pari al 20% della capacità del canale di comunicazione, che risulta essere $400Kb/s$ essendo un I2C in *fast-mode*. Questo calo è dovuto a numerosi fattori tra i quali l'overhead del protocollo e la velocità del microcontrollore.

Quindi andando a sostituire il valore stimato per la lettura di 8 bit nell'Eq.6.1 si ottiene, per l'uso dell'IMU a 9DOF:

$$T_{RX.I2C} = 3 \cdot 3 \cdot (2 \cdot T_{RX.8bit}) = 3 \cdot 3 \cdot (2 \cdot 101\mu s) = 1,8ms \quad (6.3)$$

Mentre per l'uso a 6DOF:

$$T_{RX.I2C} = 3 \cdot 2 \cdot (2 \cdot T_{RX.8bit}) = 3 \cdot 2 \cdot (2 \cdot 101\mu s) = 1,2ms \quad (6.4)$$

6.1.2 Analisi del tempo di trasmissione dei dati tramite USB

Il tempo necessario per trasmettere un pacchetto, dal modulo *microcontrollore* al modulo *App* (Cap.2.2) mediante il canale USB (Cap.5.2.2), è dato dalla seguente equazione:

$$T_{TX.USB} = \frac{dim(pack)}{goodput} \quad (6.5)$$

Per stimare il *goodput* si è programmato il microcontrollore in modo da inviare 1000 volte un pacchetto contenente un *id* (univoco e incrementale) e un contatore incrementato quando il buffer di trasmissione risulta essere ancora occupato. Per il codice si rimanda alla lettura dell'App.C.1.2. Eseguendo il test numerose volte, si è osservato che il tempo necessario per trasmettere 1000 pacchetti, per un totale di 8574 Bytes, è di circa $57ms$. Questo implica un *goodput* di $1,2Mb/s$ pari al 10% della capacità del canale di comunicazione che, essendo un USB 1.1, risulta essere $12Mb/s$. Ancora una volta questo calo di prestazioni è dovuto a numerosi fattori tra i quali l'overhead del protocollo, l'implementazione della libreria STM utilizzata e la velocità del microcontrollore.

Più complicato è il discorso riguardante la dimensione dei pacchetti da trasmettere, questi infatti sono strettamente correlati a:

- **La modalità di computazione** selezionata (Cap.5.1)
- **Il numero di sensori** utilizzati
- **La codifica** dell'informazione

Con riferimento alle possibili combinazioni dei fattori precedenti e alle rispettive strutture dei pacchetti, dettagliate nel Cap.5.1.3, si hanno le seguenti dimensioni:

1. **LCM:**

(a) $6DOF = 48 \text{ Bytes}$

(b) $9DOF = 63 \text{ Bytes}$

2. **HCM** = 21 Bytes

3. **TCM:**

(a) $6DOF = 63 \text{ Bytes}$

(b) $9DOF = 84 \text{ Bytes}$

Sostituendo questi valori e la stima del *goodput* del canale all'Eq.6.5, si ottengono le stime dei diversi tempi di trasmissione:

1. **LCM:**

(a) $6DOF: T_{TX_USB} = \frac{48 \text{ Bytes}}{1,2 \text{ Mbit/s}} = 0,32 \text{ ms}$

(b) $9DOF: T_{TX_USB} = \frac{63 \text{ Bytes}}{1,2 \text{ Mbit/s}} = 0,42 \text{ ms}$

2. **HCM**: $T_{TX_USB} = \frac{21 \text{ Bytes}}{1,2 \text{ Mbit/s}} = 0,16 \text{ ms}$

3. **TCM:**

(a) $6DOF: T_{TX_USB} = \frac{63 \text{ Bytes}}{1,2 \text{ Mbit/s}} = 0,42 \text{ ms}$

(b) $9DOF: T_{TX_USB} = \frac{84 \text{ Bytes}}{1,2 \text{ Mbit/s}} = 0,56 \text{ ms}$

6.1.3 Conclusioni analisi temporale

Sommando i tempi stimati nei paragrafi precedenti (Cap.6.1.1 e Cap.6.1.2), si ottengono le diverse finestre temporali minime affinché, il microcontrollore riesca a completare le funzioni basilari:

1. **LCM:**

(a) $6DOF: T_{min} = T_{RX_I2C} + T_{TX_USB} = 1,2 \text{ ms} + 0,32 \text{ ms} = 1,52 \text{ ms}$

(b) $9DOF: T_{min} = T_{RX_I2C} + T_{TX_USB} = 1,8 \text{ ms} + 0,42 \text{ ms} = 2,22 \text{ ms}$

2. HCM :

$$(a) \ 6DOF: T_{min} = T_{RX.I2C} + T_{TX.USB} = 1,2ms + 0,16ms = 1,36ms$$

$$(b) \ 9DOF : T_{min} = T_{RX.I2C} + T_{TX.USB} = 1,8ms + 0,16ms = 1,96ms$$

3. TCM:

$$(a) \ 6DOF : T_{min} = T_{RX.I2C} + T_{TX.USB} = 1,2ms + 0,42ms = 1,62ms$$

$$(b) \ 9DOF : T_{min} = T_{RX.I2C} + T_{TX.USB} = 1,8ms + 0,56ms = 2,36ms$$

Da notare che tali stime non tengono conto di alcune latenze legate all'esecuzione del codice all'interno del microprocessore, per questo motivo si ritiene opportuno sovrastimare le finestre temporali e sottostimare le frequenze massime, in modo da avere margini di sicurezza e stabilità maggiori.

Per le modalità *HCM* e *TCM* si deve tener conto anche del tempo di computazione della libreria *Motion FX* per la stima della dell'assetto a partire dai dati grezzi che, nel caso peggiore è dichiarato essere $3,2ms$ [22] per il microcontrollore utilizzato nell'ambito di questa tesi.

Quindi le stime aggiornate risultano essere:

1. LCM:

$$(a) \ 6DOF: T_{min} = T_{RX.I2C} + T_{TX.USB} = 1,2ms + 0,32ms = 1,52ms$$

$$(b) \ 9DOF : T_{min} = T_{RX.I2C} + T_{TX.USB} = 1,8ms + 0,42ms = 2,22ms$$

2. HCM :

$$(a) \ 6DOF: T_{min} = T_{RX.I2C} + T_{TX.USB} + T_{MotionFX} = 1,2ms + 0,16ms + 3,2ms = 4,56ms$$

$$(b) \ 9DOF : T_{min} = T_{RX.I2C} + T_{TX.USB} + T_{MotionFX} = 1,8ms + 0,16ms + 3,2ms = 5,16ms$$

3. TCM:

$$(a) \ 6DOF : T_{min} = T_{RX.I2C} + T_{TX.USB} + T_{MotionFX} = 1,2ms + 0,42ms + 3,2ms = 4,82ms$$

$$(b) \ 9DOF : T_{min} = T_{RX.I2C} + T_{TX.USB} + T_{MotionFX} = 1,8ms + 0,56ms + 3,2ms = 5,56ms$$

Che corrispondono alle seguenti frequenze di lavoro massime:

1. **LCM:**

(a) $6DOF: f = 657Hz$

(b) $9DOF : f = 450Hz$

2. **HCM :**

(a) $6DOF: f = 219Hz$

(b) $9DOF : f = 193Hz$

3. **TCM:**

(a) $6DOF : f = 207Hz$

(b) $9DOF : f = 179Hz$

Capitolo 7

Conclusioni e prospettive future

Appendice A

Quaternioni e rotazioni nello spazio

Appendice B

Filtro di Kalman

Appendice C

Script

C.1 Testing

C.1.1 Script per la stima del tempo di lettura di dati provenienti dall'IMU

Di seguito si riporta il codice utilizzato per stimare il tempo di lettura di dati provenienti dall'unità di misura inerziale utilizzando un canale I2C in fast-mode.

```
//Test I2C speed
while(1) {
    if(id<1000) {

        if(!I2C_ReadData(LSM9DS1_I2C_BADD,
            LSM9DS1_WHO_AM_I,&result,1)) {
            if( result == LSM9DS1_WHO_AM_I_VALUE)
                id++;
        }
        else
        {
            lost++;
        }

        if(id==1000) {
            t1=HAL_GetTick()-t0;
```

```
        HAL_Delay(2);
        snprintf(data, 10, "%Lu", t1);
        CDC_Transmit_FS((uint8_t *)data, strlen(data));
    }
}
```

C.1.2 Script per la stima del tempo di trasmissione di dati al modulo App

Di seguito si riporta il codice utilizzato per stimare il tempo di trasmissione sul canale USB utilizzato nella classe CDC.

```
//Test usb CDC speed
while(1)
{
    snprintf(data, 10, "%d|%d\n", id, lost);
    if(id<1000) {
        if(CDC_Transmit_FS((uint8_t *)data, strlen(data)) == USB_OK)
        {
            id++;
        }
        else{
            lost++;
        }
    }

    if(id==1000) {
        t1=HAL_GetTick()-t0;
        snprintf(data, 10, "%Lu\n", t1);
        CDC_Transmit_FS((uint8_t *)data, strlen(data));
    }
}
```


Bibliografia

- [1] Towards a Better Understanding of Context and Context-Awareness <ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf>
- [2] Sistema di posizionamento globale https://it.wikipedia.org/wiki/Sistema_di_posizionamento_globale#cite_note-1
- [3] Indoor positioning system https://en.wikipedia.org/wiki/Indoor_positioning_system
- [4] Progettazione e realizzazione di un Indoor Positioning System basato su geomagnetismo e sensor fusion http://amslaurea.unibo.it/12840/1/federico_torsello_tesi.pdf
- [5] Luca Pappalardo, "Localizzazione - Problema, Tecniche, Algoritmi - Reti mobili: Ad Hoc e di sensori", 2011, <http://didawiki.di.unipi.it/lib/exe/fetch.php/rhs/localizzazione.pdf>
- [6] Cuccado, De Franceschi, Fauri, Sartor, "Analisi di algoritmi di autolocalizzazione per reti di sensori wireless", 2007, <https://art.torvergata.it/retrieve/handle/2108/773/6945/Paolo-Sperandio-Tesi-PhD.pdf>
- [7] An adaptive indoor positioning system based on Bluetooth Low Energy RSSI <https://www.politesi.polimi.it/bitstream/10589/92284/3/NicolaCinefra770910TesiDefinitiva.pdf>
- [8] Harrop P, Raghu D. Mobile Phone Indoor Positioning Systems (IPS) and Real Time Locating Systems (RTLS) 2014-2024. Forecasts, Players, Opportunities. IDTechEx

- [9] Ugur Bekcibasi, "Increasing RSSI Localization Accuracy with Distance Reference Anchor in Wireless Sensor Networks", 2014, http://www.uni-obuda.hu/journal/Bekcibasi_Tenruh_54.pdf
- [10] Mak LC, Furukawa T. A ToA-based Approach to NLOS Localization Using Low-Frequency Sound. ACRA2006 (Auckland, New Zealand); 2006
- [11] A Simple Technique for angle of arrival measurement https://www.researchgate.net/publication/4368716_A_Simple_Technique_for_angle_of_arrival_measurement
- [12] Generalized Geometric Triangulation Algorithm for Mobile Robot Absolute Self-Localization <https://pdfs.semanticscholar.org/dee6/fb124433cac10744afd9502b165ffdec202.pdf>
- [13] Sensor fusion https://en.wikipedia.org/wiki/Sensor_fusion
- [14] Jeroen Hol "Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS" <https://www.xsens.com/wp-content/uploads/2014/pdf/Hol2011%20-%20Dissertation.pdf>
- [15] Janusz Bryzek "Principles of MEMS" <https://www.wiley.com/legacy/wileychi/hbmsd/pdfs/mm573.pdf>
- [16] MEMS Accelerometer <http://www.instrumentationtoday.com/mems-accelerometer/2011/08/>
- [17] H. Titterton and J. L. Weston. Strapdown inertial navigation technology. IEE radar, sonar, navigation and avionics series. Peter Peregrinus Ltd., Stevenage, UK, 1997. ISBN 0863413587.
- [18] Forza di Coriolis, Wikipedia https://it.wikipedia.org/wiki/Forza_di_Coriolis
- [19] The Development of Micromachined Gyroscope Structure and Circuitry Technology, Dunzhu Xia, Cheng Yu and Lun Kong
- [20] TECNICHE DI STIMA E CONTROLLO DI ASSETTO DI SATELLITI. Andrea Fagiani. <http://control.disp.uniroma2.it/carnevale/archivio/Tesi/andreaFagianiM/tesi.pdf>

-
- [21] Advanced algorithms and architectures for MEMS inertial sensor platforms in orientation tracking and in fall detection applications. Simone Sabatelli. Università di Pisa.
- [22] MotionFX middleware library in X-CUBE-MEMS1 software expansion for STM32Cube.
http://www.st.com/content/ccc/resource/technical/document/user_manual/group0/31/0e/66/39/cb/f7/4e/cd/DM00394369/files/DM00394369.pdf/jcr:content/translations/en.DM00394369.pdf